RANDOM SAMPLING ADC FOR SPARSE SPECTRUM SENSING

Patrick Maechler[*], *Norbert Felber*[*], *and Andreas Burg*[‡]

[*]Integrated Systems Laboratory, ETH Zurich
Gloriastr. 35, 8092 Zurich, Switzerland
{maechler,felber}@iis.ee.ethz.ch

[‡]Telecommunications Circuits Laboratory, EPFL
Station 11, 1015 Lausanne, Switzerland
andreas.burg@epfl.ch

ABSTRACT

Scanning large bandwidths (spectrum sensing) pushes today's analog hardware to its limits since periodic sampling at Nyquist rate with sufficient resolution is often prohibitively complex. In this paper, we consider a scenario where the signal to be acquired is sparse in the frequency domain (e.g., spectrum sensing in cognitive radio applications) and we are interested in identifying the sparse support of the signal. For this type of applications, we describe a new analog-to-digital converter (ADC) architecture that acquires unequally spaced samples based on a slope ADC, which is one of the least complex ADC architectures available. For the signal reconstruction, we employ algorithms from compressed sensing for the recovery of the dominant spectral components. The performance of the proposed design is compared to more traditional designs with comparable or higher hardware complexity.

1. INTRODUCTION

Spectrum sensing, i.e. the identification of occupied frequency bands, is a task which appears in many applications. One of them is cognitive radio (CR) [3], which allows to reuse free bands (white spaces) in the spectrum of licensed users. CR can increase spectrum utilization especially when the spectrum is only sparsely used. In order to transmit on a dynamically assigned frequency, one has to verify that no interference is caused to any licensed user which is currently transmitting. Identifying unused frequency bands by scanning large bandwidths is a challenging task when bandwidths in the GHz range are to be observed. Traditional approaches either split the signal among a number of narrow-band detectors by a bank of passband-filters or the wideband signal is sampled directly at Nyquist rate by a high-speed ADC. Both techniques result in large, expensive, and power hungry hardware implementations.

In order to reduce the effort for spectrum sensing, additional constraints on the signal must be introduced. In this paper, we assume that the observed signal is sparse in the frequency domain. Hence, in a given time window only a small number of bands (frequencies) are occupied[1]. Our goal is to find a hardware implementation that is as simple as possible, but can still reliably detect all occupied frequency bands.

A promising approach to exploit the sparsity assumption for reducing the complexity of the sampling process comes from the area of compressed sensing (CS) [2]. In essence, the corresponding recent results explain why and how signals with a sparse representation in some domain can (under certain conditions) be sampled below their Nyquist rate and how the original signal can be reconstructed. Unfortunately, straightforward periodic undersampling is not possible. Hence, new sampling hardware (analog-to-information converter) is required to realize the proper sampling process. The construction of such hardware is far from trivial since it requires new ideas and topologies that are quite different from the well studied traditional ADCs.

Prior work The related prior work that is concerned with the construction of reduced-complexity hardware for sampling and recovering of frequency-sparse signals can be divided into two approaches:

The first approach is to sample the time-domain signal in a basis that is different from the canonical basis. To this end, special analog frontends are employed to perform the projection of the incoming signal onto a random basis. For example, the *random demodulator* in [10] first multiplies the signal with a high-frequency pseudo-noise sequence, accumulates the result in an integrator, and then samples at a reduced frequency. A parallel random demodulator was proposed in [11] where the signal is projected onto multiple Bernoulli waveforms in the analog domain. Similarly, the *modulated wideband converter* (MWC) [6] uses low-pass filters instead of integrators and periodic high-frequency waveforms. In both cases, the mixing with high-frequency analog signals is practically feasible. However, the additional analog frontend also adds non-negligible area overhead and complicates the analog design, which is undesirable, especially in modern process technologies where the trend is toward shifting complexity from the analog domain to the digital signal processing.

The second approach is to exploit the fact that a signal that is sparse in the frequency domain can be reconstructed after acquiring it in the time domain using unequally spaced samples. The main difficulty lies in the fact that random sampling may occasionally lead to samples that are very closely spaced in time which requires hardware that can also handle very short sampling intervals even though the average sampling rate is well below the worst case. Possible implementation strategies for the *random sampling* approach are for example described in [4] and [9] where the authors propose two prototypes: one uses a bank of low-rate ADCs with shifted starting points, the other one utilizes capacitors to store analog values until they are sampled by a low-rate ADC. Unfortunately, both approaches require non-trivial additional custom hardware which, in the end, is partially underutilized to enforce an irregular sampling grid.

Contributions & Outline In this work, we describe a new approach to acquire signals that are sparse in the frequency domain using random sampling. Our solution comprises a

[1] In many CR settings this will be a valid assumption given the measured low spectrum utilization.

very simple sampling hardware and a slightly adapted CS reconstruction algorithm for sparse signal recovery. In particular, our ADC is based on a small modification of the well-known slope ADC and achieves a 100% hardware utilization to maximize the number of unequally-spaced samples that can be acquired without adding complexity to the sampling process. The rest of the paper is organized as follows: We first briefly review the basics of sparse signal recovery and introduce ideal random sampling with an adapted reconstruction algorithm in Sec. 2. In Sec. 3, we explain how random sampling can be implemented in a very simple and hardware-efficient way. Simulation results and a comparison to related techniques are shown in Sec. 4 and conclusions are drawn in Sec. 5.

2. RANDOM SAMPLING

CS provides a framework with reconstruction algorithms and convergence criteria that allows to reconstruct sparse or compressible signals from far fewer measurements than the dimension of the unknown signal suggests [1, 2]. A signal is called K-sparse if it can be represented by K coefficients in a given basis. If $\mathbf{x} \in \mathbb{C}^M$ is the sparse representation of this signal, only K out of its M coefficients have non-zero values. For a measurement vector $\mathbf{y} \in \mathbb{C}^N$ with potentially $N \ll M$, and the measurement matrix (or dictionary) $\mathbf{\Phi} \in \mathbb{C}^{N \times M}$, one can reconstruct $\mathbf{x}$ from $\mathbf{y} = \mathbf{\Phi x}$ if $\mathbf{\Phi}$ fulfills certain conditions on the restricted isometry property. Columns of $\mathbf{\Phi}$ are called atoms or dictionary elements and should be incoherent to fulfill the above property. With noise-free measurements, the signal $\mathbf{x}$ can be reconstructed by finding the sparsest possible solution which fulfills $\mathbf{y} = \mathbf{\Phi x}$ according to

$$\hat{\mathbf{x}} = \text{argmin}_{\mathbf{x}} ||\mathbf{x}||_0, \text{ subject to } \mathbf{y} = \mathbf{\Phi x}. \tag{1}$$

A large number of algorithms have been proposed to solve the reconstruction problem in (1). There are mainly two classes: convex optimizers, which minimize the l_1-norm instead of the l_0-norm, and greedy algorithms which approach the optimal solution through iterative procedures. Many algorithms also work for noisy measurements where the reconstruction problem is not solved with equivalence but within an error bound $||\mathbf{\Phi x} - \mathbf{y}||^2 \leq \varepsilon$.

2.1 Ideal Random Sampling

In the spectrum sensing application considered in this paper, the continuous time signal $y(t)$ is assumed to be band-limited to a frequency f with a sufficiently sparse discrete frequency domain representation $\mathbf{x}$. The measurements $\mathbf{y}$ are taken in the time domain with a time-resolution $T_0 < 1/f_N$ with $f_N = 2f$ being the Nyquist frequency. For the conventional periodic sampling, the measurement matrix $\mathbf{\Phi}$ corresponds to a full discrete Fourier transform (DFT) matrix with $N = M$.

$$y_n = y(t)|_{t=k_n T_0}, \quad k_n \in \{1, M\}, \quad k_n > k_{n-1} \tag{2}$$

$$n = 1, \ldots, N$$

$$\mathbf{\Phi}_{n,m} = \frac{1}{\sqrt{N}} \exp\left(-j2\pi \frac{(k_n - 1)(m-1)}{M}\right) \tag{3}$$

$$m = 1, 2, \ldots, M$$

By reducing the number of measurements, i.e., $N < M$, the signal $y(t)$ becomes undersampled with an effective sampling rate of $f_{\text{eff}} = N/(MT_0)$. If the sampling instances k_n are selected randomly from the grid with resolution T_0, the measurement matrix created according to (3) fulfills the criteria for reliable signal recovery with high probability provided that a sufficiently large number N of time domain samples is collected [1].

2.2 Reconstruction of Real-Valued Signals

Since we consider a *real-valued* time domain signal, the corresponding spectrum $\mathbf{x} \in \mathbb{C}^M$ is conjugate symmetric. For optimal signal reconstruction, this knowledge must be incorporated into the reconstruction algorithm. To this end, real and imaginary parts of $\mathbf{x}$ are first separated. Since x_1 and $x_{M/2+1}$ are real-valued, we end up with $M/2+1$ real values and $M/2-1$ imaginary values. The total number of degrees of freedom is thus M. Since the two real-valued components x_1 and $x_{M/2+1}$ have no practical significance, we skip them for the sake of simplifying the notation. The new real-valued vector $\bar{\mathbf{x}} \in \mathbb{R}^{M-2}$ is constructed as

$$\begin{aligned} \bar{\mathbf{x}} \quad = \quad & [\text{Re}\{x_2\}\, \text{Re}\{x_3\} \ldots \text{Re}\{x_{M/2}\} \\ & \text{Im}\{x_2\}\, \text{Im}\{x_3\} \ldots \text{Im}\{x_{M/2}\}]^T. \end{aligned} \tag{4}$$

The corresponding measurement matrix, for which $\mathbf{y} = \bar{\mathbf{\Phi}}\bar{\mathbf{x}}$, is also real-valued.

$$\bar{\mathbf{\Phi}}_{n,m} = \begin{cases} \frac{2}{\sqrt{N}} \cos\left(2\pi \frac{(k_n-1)m}{M}\right) & \text{if } 1 < m < \frac{M}{2} \\ \frac{2}{\sqrt{N}} \sin\left(2\pi \frac{(k_n-1)(\frac{3}{2}M-m)}{M}\right) & \text{if } \frac{M}{2} \leq m \leq M-2 \end{cases} \tag{5}$$

For the reconstruction, we use the well-known Compressive Sampling Matching Pursuit (CoSaMP) algorithm [8]. This iterative greedy algorithm chooses in each iteration the $2K$ best fitting atoms and then performs a least squares optimization in the sub-space spanned by these atoms. Only the K best atoms are kept and will be used as a starting point for the next iteration, where new atoms can be added and obsolete ones can be discarded.

To account for the dependency of the two elements $\bar{x}_n, \bar{x}_{M/2-1+n}$, representing the real and imaginary parts of the same discrete frequency component, we modify the original algorithm to always select the two corresponding components together. To this end, we perform the selection using the magnitude of the complex coefficient, i.e. the sum of the two squared components $\sqrt{\bar{x}_h^2 + \bar{x}_{h+M/2-1}^2}$. Thus, $2K$ component-pairs are selected in the first step and are then reduced to K component-pairs.

3. SLOPE ADC

All hardware architectures of random samplers presented so far require some operations in the analog domain before sampling. A promising hardware architecture, which comes with no additional overhead, can be developed from the traditional slope ADC, which is briefly described here. The principle of a slope ADC is very simple: A linear voltage slope is generated as a reference signal and is compared to the input voltage. A counter measures the time until the reference slope reaches the level of the signal. The reference slope is *periodically* reset to its baseline as illustrated in the upper half of Fig. 1.

The advantage of this architecture is its simplicity in hardware. Basic building blocks are only a current source,

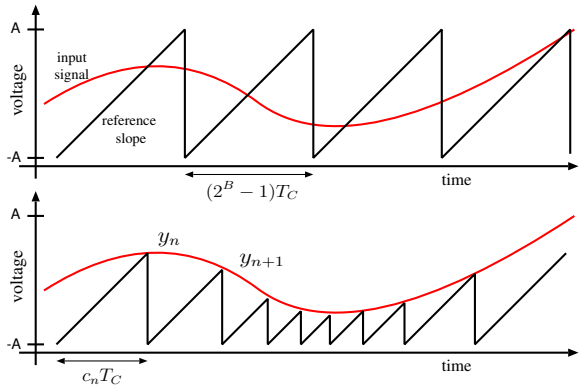

Figure 1: Slopes generated for given input signal (red) in a traditional slope ADC (top) and a RSS-ADC (bottom)

a capacitor, a comparator, and a counter (Fig. 2). Disadvantages are its slow speed, especially for high resolutions, and - in traditional systems - the unevenly spaced sampling instances. Nevertheless, this ADC architecture is getting particularly attractive for nanometer-scale CMOS converters due to a higher energy efficiency compared to flash ADCs. For example, the design in [7] achieves 1 MS/s with 9 bit resolution and 14 μW power dissipation in a 90 nm CMOS technology.

In a slope ADC, the counter clock period T_C is the smallest resolvable time. When the reference slope and the input signal intersect, the next integer multiple of T_C is recorded. In order to achieve a sampling rate of $f_S \geq f_N$ with a resolution of B bits, the clock frequency $f_C = 1/T_C$ must be set to

$$f_C = (2^B - 1) f_S.$$

The requirements on the clock frequency of the counter and the bandwidth of the comparator thus increase exponentially with the resolution B, which requires the comparator to run significantly above the intended sampling frequency f_S. In this architecture, the effects of unevenly spaced samples need to be compensated by digital postprocessing. For example by using iterative lowpass filtering [5], the sampling instances can be fit into a periodic grid again. However, this approach requires f_S to be significantly above the Nyquist rate.

3.1 Random Sampling Slope ADC (RSS-ADC)

The proposed RSS-ADC differs in two points from the conventional slope ADC. First, a more powerful digital postprocessing using CS reconstruction algorithms is applied and second, the slope reset procedure in the analog frontend is modified.

Instead of regarding the jitter around the f_S-periodic sampling grid as an imperfection which must be corrected, the uneven sampling points can be exploited when they are interpreted as randomly spaced samples. As presented in Sec. 2, random sampling is a method to enable the reconstruction of a sparse signal. As a first step, the digital postprocessing of the conventional slope ADC is therefore enhanced to deal with irregular timing and frequency sparsity. As for the random sampler in Sec. 2, CS algorithms are a proven tool to reconstruct signals under these conditions. Thus, we regard this CS-aided slope ADC as an efficient and practical implementation of a random sampler.

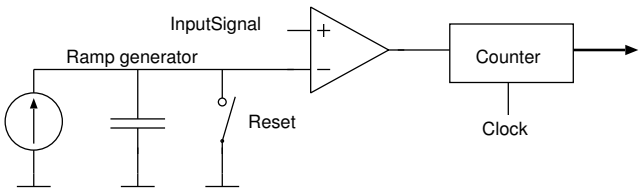

Figure 2: Schematic of a slope ADC

To further randomize the distribution of the sampling points, we propose to remove the periodicity of the reset signal. To this end, in the RSS-ADC, the reference slope is reset right after a sample has been acquired (see Fig. 1, bottom). This requires only trivial adaptations in the reset control circuitry and will not affect the requirements of the analog components. As a greatly welcome side effect, more samples can be acquired ($f_{\text{eff}} > f_S$) at no additional cost.

However, contrary to ideal random sampling, the sampling instances are signal-dependent. More samples tend to be acquired when the signal is near the starting point of the reference slope than when the signal is at the peak of the slope. The effect of this non-ideality will be studied in Sec. 4 by means of numerical simulations. Samples exceeding the range of the reference slope can be saturated or, alternatively, also be skipped. Skipping is handled well by CS algorithms but decreases the number of measurements.

3.2 CS Reconstruction

The reconstruction of the sparse frequency domain signal from the unevenly spaced measurements is done with the same algorithms as in the case of ideal random sampling described in Sec. 2. Only the measurement matrix $\tilde{\mathbf{\Phi}}$ has to be constructed such that it fits the RSS-ADC hardware.

The lowest possible sampling interval is given by the period of the counter clock $T_0 = T_C$. The sampling instances k_n are determined by the counter states c_n at which the reference slopes were reset. The series of sampling times in multiples of T_0 is then given by

$$k_n = \sum_{n'=1}^{n} c_{n'}. \tag{6}$$

New measurements are acquired until the end of the sampling window is reached at $t = MT_0$. The number of acquired measurements N is signal dependent and can not be determined in advance. The measurement matrix $\tilde{\mathbf{\Phi}}$ can be constructed using (2) and (5). Assuming ideal reference slopes, the measurements y_n are determined according to

$$y_n = c_n m + m_0. \tag{7}$$

The increment m in one clock period T_C of a reference slope raising from $-A$ to A with a resolution of B bits is given by

$$m = \frac{2A}{2^B - 2}. \tag{8}$$

The initial offset is $m_0 = -A - m/2$. In case the signal is above A or below $-A$, we set $x = \pm A \pm \frac{m}{2}$.

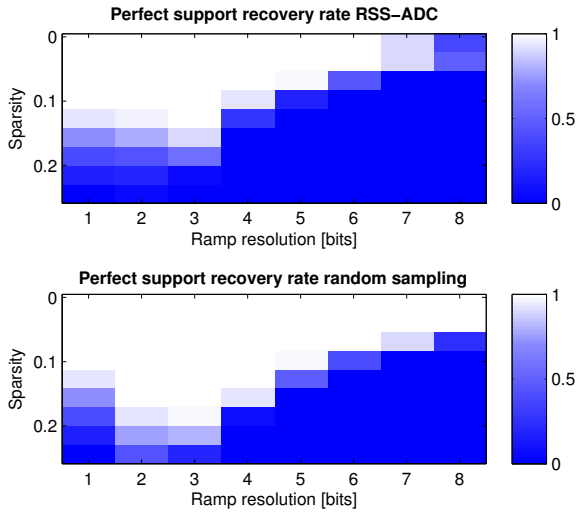

Figure 3: Perfect support recovery rate of the RSS-ADC compared to random sampling for varying sparsity and ramp resolution

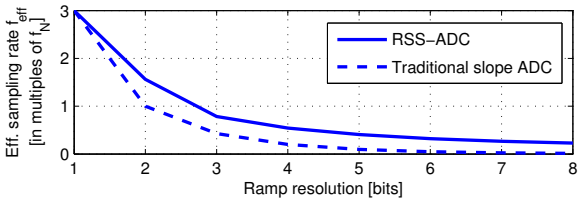

Figure 4: Effective sampling rate of the new RSS-ADC and the traditional slope ADC

4. SIMULATIONS

In this section, the performance of the new RSS-ADC is evaluated and compared to other sampling schemes. In all simulations, a signal sparse in the discrete Fourier domain is generated by randomly choosing frequency bins and assigning random phases and amplitudes. The phase is chosen uniformly at random while the amplitudes are normal distributed with a high mean value to prevent too small components. In our simulations we set the mean amplitude value to 10 times its standard deviation. The signal is then transformed into an oversampled time domain representation. The perfect support recovery rate is used as a quality metric. It is defined by the ratio of successful scans to the total number of scans in a Monte Carlo simulation. A scan is called successful when all the occupied frequency bins were identified correctly.

4.1 Resolution/Speed Trade-Off

Fig. 3 shows the perfect support recovery rate of a RSS-ADC with varying ramp resolution (B) in bits and varying sparsity (K/M) while f_C and the comparator bandwidth are fixed. Thus, all the systems simulated for this plot require the same hardware complexity. The counter clock frequency is set to three times the Nyquist frequency ($f_C = 3f_N$) and the number of discrete frequencies is set to $M = 512$. By sweeping the ramp resolution (number of bits B), not only the quantization granularity is changed but also the effective sampling

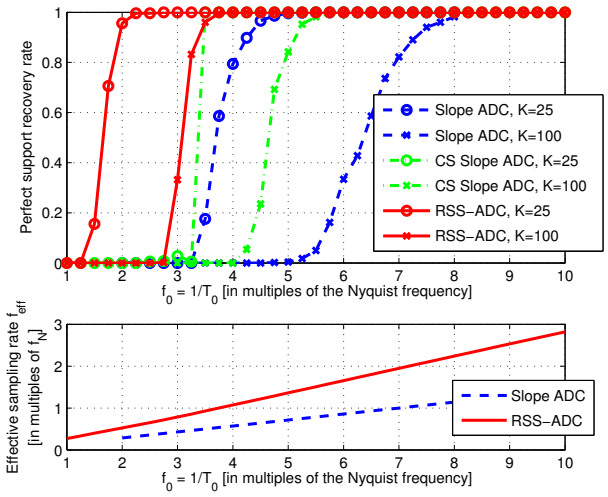

Figure 5: Comparison of traditional to CS-aided slope ADC and RSS-ADC with fixed sparsity. Top: perfect support recovery rate. Bottom: effective sampling rate

rate. The slope m gets less steep as the number of bits B increases (8) and thus it takes longer to acquire a sample. This leads to fewer but more precise measurements. The top graph in Fig. 3 depicts the performance of the new RSS-ADC.

The simulations show that with a resolution of $B = 3$ bits, the best support recovery rate is achieved. While stronger undersampling of a signal is possible with higher ramp resolution, this is not the main goal of this architecture; the efficient utilization of the hardware resources is far more important. Since the hardware costs remain constant and the performance is getting worse, higher resolution with stronger undersampling is not desired in the RSS-ADC.

4.2 Comparison to Random Sampling

The bottom graph in Fig. 3 shows the support recovery rate of an ADC whose samples are perfectly randomly distributed. The number of samples is set to the same number as acquired by the slope ADC that operates at the indicated ramp resolution. Thus, both ADCs have the same effective sampling rate f_{eff}. The random samples are quantized with the given number of bits. By comparing the two plots, the degradation of the support recovery rate due to the signal-dependent sampling introduced by the RSS-ADC can be seen.

4.3 Comparison to Conventional Slope ADC

Fig. 4 shows the average number of samples acquired by the conventional and the new RSS-ADC. At $B = 3$ bits and $f_C = 3f_N$, the proposed ADC shows an average sampling rate of more than 20% below the Nyquist frequency. Using the conventional slope ADC with $B = 3$, only an undersampling with factor $f_C/(2^B - 1) = 0.43f_N$ could be obtained and reconstruction would be more difficult.

A support recovery performance comparison of the traditional slope ADC to the CS-aided slope ADC and to the RSS-ADC is shown in the upper part of Fig. 5. The most important parameter determining the system complexity and performance is the clock frequency f_C. By sweeping f_C we compare the three ADCs with the same analog hardware

complexity for a given f_C. For all ADCs, we set a quantization accuracy of $B = 3$ bits and $M = 512$. The comparison is done at two different sparsity levels of $K = 25\,(\approx 5\%$ of $M)$ and $K = 100\,(\approx 20\%$ of $M)$.

In the traditional converter, the iterative low-pass filtering method described in [5] is applied to the nonuniform samples in order to recover the band-limited signal. An FFT is then applied and the frequency components showing the K highest magnitudes are selected. As an intermediate step, also the performance of a CS-aided slope ADC with full slopes but a CS reconstruction algorithm instead of low-pass filters is plotted here. This comparison shows the favorable effect of CS-aided digital postprocessing without any adaptations of the analog frontend. The last pair of curves shows the reconstruction performance of the proposed RSS-ADC including fast slope reset and CS-aided reconstruction as a function of f_C. As illustrated at the bottom of Fig. 5, the RSS-ADC increases the effective sampling rate compared to the conventional slope ADC hardware. As shown in the upper part of Fig. 5, this increased number of measurements allows to further reduce the clock frequency f_C required for successful support recovery.

5. CONCLUSION

A modified slope ADC, which delivers irregular samples at a high rate, was presented for sparse spectrum sensing. The new random sampling slope ADC (RSS-ADC) uses a compressed sensing reconstruction algorithm to take advantage of the inherently irregular sampling process and resets the slope right after the signal detection to increase both randomness and sampling rate. For the CS reconstruction algorithm, a version slightly adapted for real-valued signals of the CoSaMP algorithm is used. The proposed RSS-ADC increases the spectrum sensing performance compared to the conventional slope ADC at no extra cost in analog hardware.

An issue not addressed in this paper is the recovery of frequencies which are not on the discrete grid as assumed by our model. Overcomplete dictionaries or subspace-based recovery methods are possible remedies and will be discussed in future work.

Acknowledgment

Financial support for this work has been provided by the Hasler Foundation.

REFERENCES

[1] E. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory*, 52(2):489–509, Feb. 2006.

[2] D. Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, April 2006.

[3] S. Haykin. Cognitive radio: brain-empowered wireless communications. *IEEE Journal on Selected Areas in Communications*, 23(2):201–220, Feb. 2005.

[4] J. Laska, S. Kirolos, Y. Massoud, R. Baraniuk, A. Gilbert, M. Iwen, and M. Strauss. Random sampling for analog-to-information conversion of wideband signals. In *IEEE Dallas/CAS Workshop on Design, Applications, Integration and Software*, pages 119 –122, Oct. 2006.

[5] F. Marvasti, M. Analoui, and M. Gamshadzahi. Recovery of signals from nonuniform samples using iterative methods. *IEEE Transactions on Signal Processing*, 39(4):872 –878, April 1991.

[6] M. Mishali and Y. Eldar. From theory to practice: Sub-Nyquist sampling of sparse wideband analog signals. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):375–391, 2010.

[7] S. Naraghi, M. Courcy, and M. Flynn. A 9-bit, 14 μW and 0.06 mm^2 pulse position modulation ADC in 90 nm digital CMOS. *IEEE Journal of Solid-State Circuits*, 45(9), 2010.

[8] D. Needell and J. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, May 2009.

[9] T. Ragheb, S. Kirolos, J. Laska, A. Gilbert, M. Strauss, R. Baraniuk, and Y. Massoud. Implementation models for analog-to-information conversion via random sampling. In *50th Midwest Symposium on Circuits and Systems (MWSCAS)*, pages 325–328, Aug. 2007.

[10] J. Tropp, J. Laska, M. Duarte, J. Romberg, and R. Baraniuk. Beyond Nyquist: Efficient sampling of sparse bandlimited signals. *IEEE Trans. Inf. Theory*, 56(1):520–544, Jan. 2010.

[11] Z. Yu, S. Hoyos, and B. Sadler. Mixed-signal parallel compressed sensing and reception for cognitive radio. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3861–3864, 2008.